

In the Claims:

Please amend the claims as follows:

1. (currently amended) A method to revalidate a compiler and a compiler execution environment intended for compilation of a user-written program for safety control in an industrial control system after ~~it has been used~~ use of the compiler and the compiler execution environment, the method comprising:

compiling a test program a first time which test program is defined in a control language;
validating the compiler and the compiler execution environment by verifying that the test program executes correctly;

generating a first software means derived from the compiled test program intended for later comparison purposes;

compiling the test program a second time after the compilation of a user-written program;
generating a second software means intended for a comparison based on the second compilation of the test program;

comparing the first software means with the second software means; wherein the compiler and the compiler execution environment ~~are~~ is revalidated for any errors introduced between the first and the second compilation; and

enabling, provided that the revalidation indicates no errors in the compiler and the compiler execution environment, the user-written program to execute in a device with safety features for control of real world entities.

2. (currently amended) The method according to claim 1, wherein the comparing ~~step~~ is performed in the same workstation or general-purpose computer as that in which the compiler is executing.

3. (previously amended) The method according to claim 1, wherein the software means is a check-sum or a code for cyclic redundancy check.

4. (currently amended) The method according to claim 3, wherein the comparing ~~step~~ is performed in the device with safety features.

5. (currently amended) The method according to claim 4, wherein the comparing ~~step~~ further comprises ~~an additional step of~~ downloading a variable that changes over time, which is downloaded in the same message as the check-sum or code to the device, where the variable that changes over time is used to achieve a change in the message.

6. (previously amended) The method according to claim 1, wherein the test program is defined in a control language derived from the standard IEC 6-1131.

7. (currently amended) A computer program product, comprising:
computer readable medium; and
computer program instructions recorded on the computer readable medium and
executable by a processor for carrying out a ~~containing software code means loadable into the~~
~~internal memory of a general-purpose computer or workstation and/or a device, wherein said~~

~~computer program product has means to execute a computer implemented step of compiling the test program a second time, a computer implemented step of generating a second software means, a computer implemented step of comparing the first software means with the second software means and a computer implemented step of enabling the user written program to execute in the device, all steps according to claim 1~~ method to revalidate a compiler and a compiler execution environment intended for compilation of a user-written program for safety control in an industrial control system after use of the compiler and the compiler execution environment, the method comprising:

compiling a test program a first time which test program is defined in a control language;
validating the compiler and the compiler execution environment by verifying that the test program executes correctly;

generating a first software means derived from the compiled test program intended for later comparison purposes;

compiling the test program a second time after the compilation of a user-written program;
generating a second software means intended for a comparison based on the second compilation of the test program;

comparing the first software means with the second software means; wherein the compiler and the compiler execution environment are revalidated for any errors introduced between the first and the second compilation; and

enabling, provided that the revalidation indicates no errors in the compiler and the compiler execution environment, the user-written program to execute in a device with safety features for control of real world entities.

8. (cancelled)

9. (cancelled)